

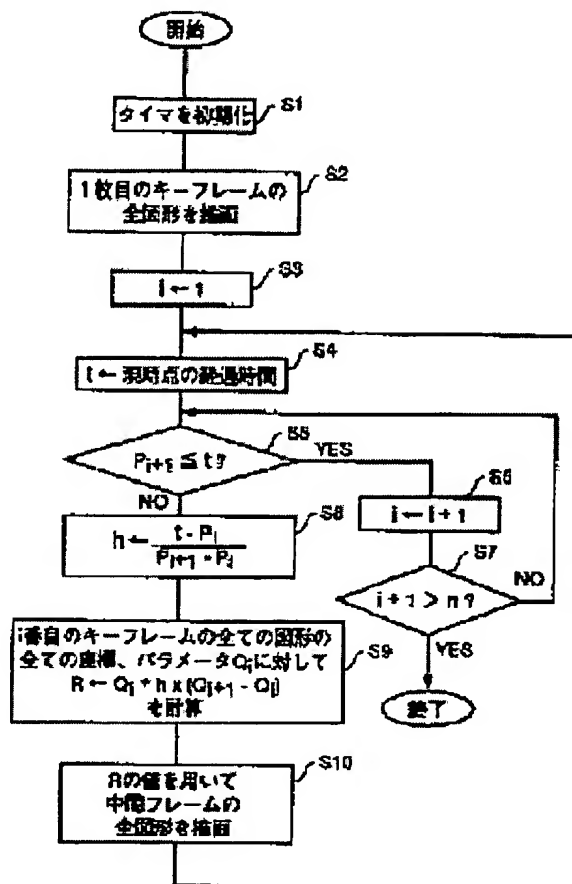
ANIMATION DISPLAY METHOD AND COMPUTER READABLE RECORDING MEDIUM RECORDING ANIMATION DISPLAY PROGRAM

Patent number: JP10275244
Publication date: 1998-10-13
Inventor: NISHIHATA MINORU
Applicant: SHARP KK
Classification:
- international: G06T13/00; G09G5/00; G09G5/36; G09G5/38;
G06T13/00; G09G5/00; G09G5/36; G09G5/38; (IPC1-7): G06T13/00; G09G5/00; G09G5/36; G09G5/38
- european:
Application number: JP19970359041 19971226
Priority number(s): JP19970359041 19971226; JP19970015413 19970129

Report a data error here

Abstract of JP10275244

PROBLEM TO BE SOLVED: To reproduce a computer animation of a key frame interpolation system for a fixed time regardless of the CPU speed and the drawing value per frame. **SOLUTION:** The current display time is decided based on plural pieces of key frame information including the information that specifies the display time (S4), and the key frame information preceding and following the current display time are selected (S5). A prescribed interpolating operation is performed between the graphic information corresponding to each other which are included in those selected key frame information, and the graphic information shown by the display time is produced (S8, S9). Based on this graphic information, a frame is drawn (S10). It's desirable to start the drawing of the next frame after the first frame is drawn.



Data supplied from the [esp@cenet](http://www.esp@cenet.com) database - Worldwide

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平10-275244

(43) 公開日 平成10年(1998)10月13日

(51) Int.Cl.⁶

識別記号

F I

G 0 6 T 13/00

C 0 6 F 15/62

3 4 0 D

G 0 9 G 5/00

5 5 0

C 0 9 G 5/00

5 5 0 B

5/36

5 1 0

5/36

5 1 0 M

5/38

5/38

B

審査請求 未請求 請求項の数15 O L (全 18 頁)

(21) 出願番号 特願平9-359041

(22) 出願日 平成9年(1997)12月26日

(31) 優先権主張番号 特願平9-15413

(32) 優先日 平9(1997)1月29日

(33) 優先権主張国 日本 (J P)

(71) 出願人 000005049

シャープ株式会社

大阪府大阪市阿倍野区長池町22番22号

(72) 発明者 西畑 実

大阪府大阪市阿倍野区長池町22番22号 シ

ャープ株式会社内

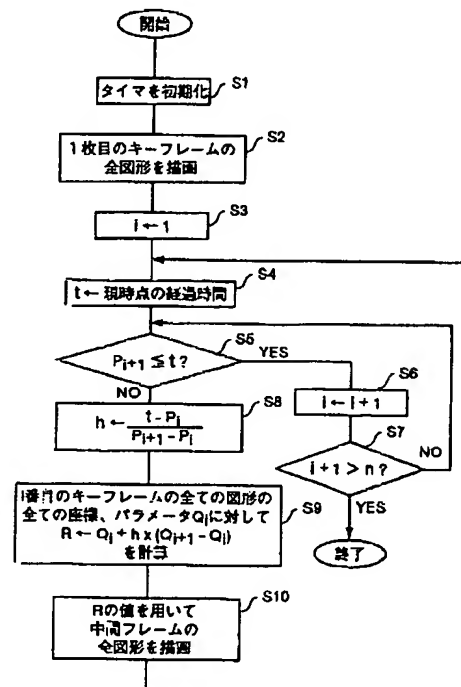
(74) 代理人 弁理士 梅田 勝

(54) 【発明の名称】 アニメーション表示方法、及びアニメーション表示プログラムを記録したコンピュータ読み取り可能な記録媒体

(57) 【要約】

【課題】 キーフレーム補間方式のコンピュータアニメーションにおいて、CPU速度や1フレーム当りの描画量に依存せずに、一定時間の再生をできるようにする。

【解決手段】 表示時刻を特定する情報を有する複数のキーフレーム情報を準備し、現在の表示時刻を得て (S4)、この表示時刻の前後のキーフレーム情報を選択し (S5)、これらキーフレーム情報に含まれた対応の図形情報の間で所定の補間演算を行って、この表示時刻で表示される図形情報を作成し (S8、S9)、この図形情報に基づきフレームを描画する (S10)。1つのフレームの描画処理が完了すると、次のフレームの描画処理を開始するようにするのが好ましい。



【特許請求の範囲】

【請求項1】 コンピュータプログラムを用いて図形のアニメーションを表示装置上に表示するためのアニメーション表示方法であって、

各々が、表示時刻を特定するための情報と、当該表示時刻に表示フレーム内に表示されるべき図形の特徴を特定するための図形情報とを含む、複数のキーフレーム情報をメモリ内に準備するステップと、

現在の表示時刻を得るステップと、

前記複数のキーフレーム情報のうち、得られた表示時刻の前後の表示時刻をそれぞれ有する少なくとも2つのキーフレーム情報を選択するステップと、

前記少なくとも2つのキーフレーム情報にそれぞれ含まれた対応の図形情報の間で所定の補間計算を行なうことにより、当該得られた表示時刻において表示されるべき図形情報を作成するステップと、

作成された図形情報に基づいて、当該得られた表示時刻におけるフレームを表示装置上に表示するステップと、前記現在の表示時刻を得るステップと、前記選択するステップと、前記作成するステップと、前記表示するステップとを、所定時刻が経過するまで繰返すステップとを含む、アニメーション表示方法。

【請求項2】 前記複数のキーフレーム情報は、先頭に表示されるべきフレームに関する先頭キーフレーム情報を含み、

前記方法はさらに、

アニメーション表示の開始時に、前記先頭キーフレーム情報に基づいて先頭フレームを表示するステップを含み、

前記現在の表示時刻を得るステップは、

1つのフレームの描画処理が完了したことに応答して、タイマを参照し、その時点でのタイマの値を表示時刻として得るステップを含む、請求項1に記載のアニメーション表示方法。

【請求項3】 前記図形情報が、図形の特徴点の座標情報を含む、請求項1または2に記載のアニメーション表示方法。

【請求項4】 前記図形情報が、図形の色彩情報を含む、請求項1または2に記載のアニメーション表示方法。

【請求項5】 前記図形情報が、図形の透明度の情報を含む、請求項1または2に記載のアニメーション表示方法。

【請求項6】 前記図形情報が、図形を構成する線の太さの情報を含む、請求項1または2に記載のアニメーション表示方法。

【請求項7】 前記図形情報が、図形を構成する線の、描画部分と非描画部分との比率の情報を含む、請求項1または2に記載のアニメーション表示方法。

【請求項8】 表示されるべき図形が文字であり、

前記図形情報が、文字フォントの大きさの情報を含む、請求項1または2に記載のアニメーション表示方法。

【請求項9】 表示されるべき図形が複数の図形を含む図形群であり、

前記図形情報が、図形群の基準点の座標を含む、請求項1または2に記載のアニメーション表示方法。

【請求項10】 表示されるべき図形が複数の図形を含む図形群であり、

前記図形情報が、図形群の表示上の縮尺比率を含む、請求項1または2に記載のアニメーション表示方法。

【請求項11】 前記作成するステップは、前記少なくとも2つのキーフレーム情報にそれぞれ含まれる対応の図形情報の間で所定の線形計算による補間計算を行なうステップと、

前記補間計算により得られた図形情報を、予め定められた変換方法により変換するステップを含む、請求項1または2に記載のアニメーション表示方法。

【請求項12】 アニメーションの表示の開始に先立って、前記図形情報を変換するためのテーブルを準備するステップをさらに含み、

前記変換するステップは、前記補間計算を行なうステップで作成された図形情報に対応する変換後の図形情報を前記テーブルからルックアップするステップを含む、請求項11に記載のアニメーション表示方法。

【請求項13】 前記変換するステップは、前記補間計算を行なうステップで作成された図形情報を所定の関数の変数に代入し、前記所定の関数の値を変換後の図形情報とするステップを含む、請求項11に記載のアニメーション表示方法。

【請求項14】 前記テーブルからルックアップするステップは、前記変換するためのテーブルは、識別するための番号を持ち、前記図形情報は、前記テーブルに対応づけるための番号を持つと共に、あらかじめ図形ごとに個別に対応づけられたテーブルからルックアップするステップを含む、請求項12に記載のアニメーション表示方法。

【請求項15】 コンピュータプログラムを用いて図形のアニメーションを表示装置上に表示するためのアニメーション表示プログラムを記録した記録媒体であって、各々が、表示時刻を特定するための情報と、当該表示時刻に表示フレーム内に表示されるべき図形の特徴を特定するための図形情報とを含む、複数のキーフレーム情報をメモリ内に準備するステップと、現在の表示時刻を得るステップと、前記複数のキーフレーム情報のうち、得られた表示時刻の前後の表示時刻をそれぞれ有する少なくとも2つのキーフレーム情報を選択するステップと、前記少なくとも2つのキーフレーム情報にそれぞれ含まれた対応の図形情報の間で所定の補間計算を行なうことにより、当該得られた表示時刻において表示されるべき

図形情報を作成するステップと、作成された図形情報に基づいて、当該得られた表示時刻におけるフレームを表示装置上に表示するステップと、前記現在の表示時刻を得るステップと、前記選択するステップと、前記作成するステップと、前記表示するステップとを、所定時刻が経過するまで繰返すステップとを、実現するためのアニメーション表示プログラムを記録したコンピュータ読み取り可能な記録媒体。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、コンピュータを用いて図形のアニメーションを再生するアニメーション再生方法に関し、特に、時系列内のキーとなるフレーム（キーフレーム）の画像をもとにして、キーフレーム間の画像を実時間で生成して再生するアニメーションの再生方法に関する。

【0002】

【従来の技術】コンピュータを用いたアニメーションの再生方法として、時系列中のすべてのフレームの画像ではなく、キーとなる時刻におけるフレームの画像の座標データをもとにして、キーフレーム間のフレームを生成してアニメーションを再生する方法がある。

【0003】図19を参照して、特開昭60-191366号公報において提案された従来の方法について説明する。この方法では、アニメーションのすべてのフレームの画像を記憶しておくのではなく、特定のキーフレームの画像のみを予め記憶しておく。図19に示す図形362および364がそうしたキーフレームの図形であり、これらが互いに時間軸上で隣接したキーフレームに属する図形であるものとする。特開昭60-191366号公報では、この隣接するキーフレームの図形362および364の間の時間を n 等分し、中間フレーム366、368および370（ $n=3$ と想定する。）の画像の中の対応する各点を、キーフレーム内の対応する画像図形362および364の間での線形補間によって求める。こうして、図19において矢印で示される方向に時間軸が進むものとすれば、まず図形362を含むキーフレームが、続いて中間の図形366、368および370をそれぞれ含む3つの中間フレームが、最後に図形364を含むキーフレームが表示されてアニメーションが再生されることになる。図形364を含むキーフレームを表示した後は、このキーフレームと、この次のキーフレームとの間で同様の補間計算を行ってアニメーション再生が行なわれる。

【0004】

【発明が解決しようとする課題】このようなアニメーション再生方法は、アニメーション再生のために必要とされるデータ量が、すべてのフレームの情報を持つ場合と比較して遙かに少なくすむという利点がある。従来はこのようなアニメーション再生方法を実現するのは困難

であったが、近年のCPU（中央演算処理装置）の処理能力の向上により、実現されようとしている。しかし、この方法では中間フレームをCPUが演算により生成しているために、次のような問題点がある。

【0005】図20を参照して、3つのフレーム382～386が、上述した従来の方法における中間フレームに相当する場合を想定する。各フレームについて、描画に要する時間を0.1秒とする。すなわちこの3つのフレームを再生するには、合計で0.3秒を必要とする。このアニメーションを、音声と並行として実時間で再生しようすると次のような問題が生ずる。

【0006】第1の問題点は、アニメーション作成時の想定よりも処理速度の遅いCPUでこのアニメーションを再生しようとした場合に生ずる。この場合には、図21に示すように、1フレームを描画するのに要する時間が0.1秒より長く、たとえば0.12秒を要する。そのためフレーム392～396を順に再生していくと、再生時間が想定よりも長くなることになり、しかも最初に想定された再生時間との遅れが徐々に大きくなる。一方で音声は、最初に想定された時間で再生される。結果として、表示される図形の動きと音声とのずれが次第に大きくなる。たとえば図21に示されるようにフレーム392、394および396と進むに従って、画面に再生される人物の口の形と音声とが合わないようになってしまう。

【0007】第2の問題点は、第1の問題点とは逆に、アニメーション作成時の想定よりも処理能力の高いCPUでこのアニメーションを再生しようとした場合に生ずる。この場合図22に示されるように、フレーム402、404および406の各々を描画するのに要する時間が0.1秒よりも短く、たとえば0.06秒しか要しないことがあり得る。そのためアニメーションの再生時間が、アニメーション作成時の想定よりも短くなる。音声の再生速度が一定であるとすれば、図22に示される場合にも、動きと音声とのずれが生じる。この場合にはアニメーションの動きの方が音声よりも早くなる。

【0008】以上述べたのはCPUの処理能力の相違により生ずる問題であった。しかしこのような問題は、各フレームごとに必要とされる描画量の変動によっても生じる。たとえば図23を参照して、フレーム412、414および416と、時間軸上を進むに従って、描画される図形が大きくなる場合を考える。こうした場合、図形が大きくなるほど描画に要する時間が長くなる。たとえばフレーム412の描画には0.1秒、フレーム414の描画には0.11秒、およびフレーム416の描画には0.12秒を要するものとする。この場合にも、フレーム414では0.01秒の遅れが生じ、フレーム416では0.01+0.02=0.03秒の描画の遅れが生じる。したがってこの場合にも、各フレームの再生タイミングが音声の再生タイミングとずれてしまう。

【0009】このように音声と画面との同期に関する問題を解決するために、先頭フレームを基準として、各フレームの描画処理の開始時刻を決めておくという方式も考えられる。しかしこの方式では次のような問題点が生ずる。この場合にも図20に示される例を想定する。

【0010】まず図24を参照して、CPUの処理能力が、想定されたものよりも低い場合を考える。この場合、たとえば3つのフレーム422、424および426の各々について割当てられた描画時間は0.1秒と定まっている。CPUの処理速度が遅いと、この与えられた描画時間中にすべての描画を完了することができない場合がある。すると、図24のフレーム422、424または426に示されるように、各フレームが不完全なまま次のフレームの描画が始まってしまうという問題が生じる。

【0011】逆にCPUの処理能力が想定された処理能力よりも高い場合には次のような問題が生ずる。図25を参照して、各フレーム432、434および436を処理能力の高いCPUで再生すると、たとえば各フレームの描画に要する時間が0.06秒しか必要としないことになる。各フレームの描画に割当てられた時間は0.1秒ずつである。したがって各フレームについて、描画処理が終了した後0.04秒の休止時間が生じることになる。この間にはCPUは遊んでしまう。CPUの処理能力を最大限に生かせば、図25に示されるような3枚のフレーム432、434および436だけではなく、図25に示されるように、たとえば5枚のフレーム442、444、446、448および450を再生することが可能なはずである。単位時間中に再生されるフレーム数が多いほどアニメーションの動きが滑らかとなることから、図25に示されるような休止時間の発生をできるだけ避け、図26に示されるような再生が行なえるようにした方が好ましい。

【0012】また、このようなアニメーションの再生で処理すべき対象としては、図形の位置および大きさだけではなく、たとえば図形の色彩、透明度、図形を構成する線の太さ、たとえば図形を構成する線が破線である場合の、描画部分と非描画部分との比率、文字を表示する場合のフォントサイズなどについても適用できれば好ましい。また、単一の図形だけではなく、一群の基本図形からなる図形群の移動についても同様の処理を行なえると望ましい。これはグループの表示上の縮尺比率についても同様である。さらに、キーフレームから中間フレームを生成する際に、単純な動きだけではなく時間の進行に対して特定の変化を持たせたアニメーション再生を行なえるようにできれば、より好ましい。

【0013】それゆえに請求項1に記載の発明および請求項2に記載の発明の目的は、アニメーションの再生速度を一定に保ちつつ、可能な限り多いフレーム数で、キーフレーム間の中間フレームを生成してアニメーション

を再生することができるアニメーション再生方法を提供することである。

【0014】請求項3～10に記載の発明の目的は、請求項1または2に記載の発明の目的に加えて、それぞれ図形の特徴点の時系列変化、図形の色彩の時系列変化、図形の透明度の時系列変化、図形を構成する線の太さの時系列変化、図形を構成する線の、描画部分と非描画部分との比率の時系列変化、図形が文字である場合のフォントサイズの時系列変化、図形が、図形群である場合に、その図形群の基準点の時系列変化、同じく図形群の表示上の縮尺比率の時系列変化、をアニメーション表示することができるアニメーション再生方法を提供することである。

【0015】請求項11～13に記載の発明の目的は、請求項1または2に記載の発明の目的に加えて、アニメーションの動きに変化を与えることができるアニメーション再生方法を提供することである。

【0016】

【課題を解決するための手段】請求項1に記載の発明に係るアニメーション表示方法は、コンピュータプログラムを用いて図形のアニメーションを表示装置上に表示するためのものであって、各々が、表示時刻を特定するための情報と、当該表示時刻に表示フレーム内に表示されるべき図形の特徴を特定するための図形情報とを含む、複数のキーフレーム情報をメモリ内に準備するステップと、現在の表示時刻を得るステップと、上記した複数のキーフレーム情報のうち、得られた表示時刻の前後の表示時刻をそれぞれ有する少なくとも2つのキーフレーム情報を選択するステップと、この少なくとも2つのキーフレーム情報にそれぞれ含まれる対応の図形情報の間で所定の補間計算を行なうことにより、当該得られた表示時刻において表示されるべき図形情報を作成するステップと、作成された図形情報に基づいて、当該得られた表示時刻におけるフレームを描画するステップと、上記した現在の表示時刻を得るステップと、選択するステップと、作成するステップと、表示するステップとを、所定時刻が経過するまで繰返すステップとを含んでいる。

【0017】予め、複数のキーフレーム情報がメモリ内に準備された後、以下の繰返しが行なわれる。まず現在の表示時刻を得る。そしてこの前後の2つのキーフレーム情報を選択する。これら2つのキーフレーム情報に含まれる対応の図形情報の間で所定の補間演算を行なって、得られた表示時刻において表示されるべき図形情報を作成し描画する。1フレームをすべて描画しなければ次のフレームの図形情報の作成を行なわないので、各フレームを完全に描画することができる。また予め表示時刻が定められているわけではないので、処理能力の高いCPUの場合には多数の中間フレームを、処理能力の低いCPUでは少数の中間フレームを作成することになり、CPUの能力に応じたアニメーション表示を行なう

ことができる。

【0018】請求項2に記載の発明に係るアニメーション表示方法は、請求項1に記載の発明の構成に加えて、複数のキーフレーム情報が、先頭に表示されるべきフレームに関する先頭キーフレーム情報を含んでいる。そしてこのアニメーション表示方法はさらに、アニメーション表示の開始時に、先頭キーフレーム情報に基づいて先頭のフレームを表示するステップを含む。また現在の表示時刻を得るステップは、1つのフレームの描画処理が完了したことに応じてタイマを参照し、その時点でのタイマの値を表示時刻として得るステップを含んでいる。

【0019】先頭のフレームの描画処理が完了すると、その完了に応じて次のフレームの表示時刻が得られる。この表示時刻でのフレームの描画が完了すると、その完了に応じてさらに次の表示時刻が得られる。このようにして次々とあるフレームの描画処理が完了するとその完了の時点の表示時刻でのフレームの作成と描画が繰返し行なわれるので、遊び時間がなく、CPUの能力を最大限に利用することができる。

【0020】請求項3～請求項10に記載の発明は、請求項1または2に記載の発明の構成に加えて、図形情報が、図形の特徴点の座標情報、図形の色彩情報、図形の透明度の情報、図形を構成する線の太さの情報、図形を構成する線の、描画部分と非描画部分との比率の情報、文字フォントの大きさ、図形群の基準点の座標、図形群の表示上の縮尺比率をそれぞれ含んでいる。こうした図形情報の間での補間演算で中間フレームでの図形情報を作成し描画することにより、図形の変形または移動、図形の色彩の変化、図形の透明度の変化、図形の線の太さ、図形を構成する線の破線の態様、文字フォントの大きさ、図形群の移動、図形群の表示上の縮尺比率の変化を含んだアニメーションを作成することができる。

【0021】請求項11に記載の発明に係るアニメーション表示方法は、請求項1または2に記載の発明の構成に加えて、作成するステップが、2つのキーフレーム情報にそれぞれ含まれる対応の図形情報の間で所定の線形計算により補間計算を行なうステップと、この補間計算により得られた図形情報を、予め定められた変換方法により変換するステップとを含む。

【0022】一旦線形計算による補間計算で生成された図形情報にさらに変換を行なうことで、たとえば加速度を伴った動きなど、時間に対して非直線の動きを表現することが可能となる。すなわち、アニメーションにおいて、時間軸上で変化をつけることが可能になる。

【0023】請求項12に記載の発明に係るアニメーション表示方法は、請求項11に記載の発明の構成に加えて、アニメーションの表示の開始に先立って、図形情報を変換するためのテーブルを準備するステップをさらに含んでいる。そして変換するステップは、作成するス

テップで作成された図形情報に対応する変換後の図形情報を、上記したテーブルからルックアップするステップを含む。

【0024】このように、一旦補間計算により作成された図形情報を、テーブルを用いて変換し変換後の図形情報によってアニメーションを表示することで、時間軸上の恣意的な変化をアニメーション上で実現することができる。

【0025】請求項13に記載の発明に係るアニメーション表示方法は、請求項11に記載の発明の構成に加えて、変換するステップが、作成するステップで作成された図形情報を所定の関数の変数に代入し、この関数の値を変換後の図形情報とするステップを含んでいる。

【0026】線形計算による補間計算で得られた図形情報を、さらに所定の関数の変数に代入して変換するので、線形でない動きをもアニメーション上で実現することができる。さらにそのために必要なデータ量は少なく済むという特徴がある。

【0027】請求項14に記載の発明に係るアニメーション表示方法は、請求項12に記載の発明の構成に加えて、前記テーブルからルックアップするステップは、前記変換するためのテーブルは、識別するための番号を持ち、前記図形情報は、前記テーブルに対応づけるための番号を持つと共に、あらかじめ図形ごとに個別に対応づけられたテーブルからルックアップするステップを含んでいる。

【0028】変換テーブルを複数個準備し、それぞれの図形情報に付加されたテーブル番号によって、それぞれ異なる変換テーブルからルックアップするステップを含んでいる。これにより、2つだけのキーフレーム間で、それぞれの図形に、より複雑な動きを表現させることが可能となる。さらにそのために多数のキーフレーム作成の手間を軽減し、必要なデータ量は少なくて済むという特徴がある。

【0029】請求項15に記載のアニメーション表示プログラムを記録したコンピュータ読み取り可能な記録媒体は、コンピュータプログラムを用いて図形のアニメーションを表示装置上に表示するためのアニメーション表示プログラムを記録した記録媒体であって、各々が、表示時刻を特定するための情報と、当該表示時刻に表示フレーム内に表示されるべき図形の特徴を特定するための図形情報とを含む、複数のキーフレーム情報をメモリ内に準備するステップと、現在の表示時刻を得るステップと、前記複数のキーフレーム情報のうち、得られた表示時刻の前後の表示時刻をそれぞれ有する少なくとも2つのキーフレーム情報を選択するステップと、前記少なくとも2つのキーフレーム情報にそれぞれ含まれた対応の図形情報の間で所定の補間計算を行なうことにより、当該得られた表示時刻において表示されるべき図形情報を作成するステップと、作成された図形情報に基づいて、

当該得られた表示時刻におけるフレームを表示装置上に表示するステップと、前記現在の表示時刻を得るステップと、前記選択するステップと、前記作成するステップと、前記表示するステップとを、所定時刻が経過するまで繰返すステップとを實現するためのアニメーション表示プログラムを記録したコンピュータ読み取り可能な記録媒体であることを特徴とする。

【0030】

【発明の実施の形態】図1を参照して、本願発明のアニメーション表示方法はコンピュータ30により實現される。コンピュータ30は、所定のプログラムを実行することにより本願発明の各ステップの処理を實現するためのCPU32と、CPU32がアニメーション画像を書込むグラフィックメモリ36と、グラフィックメモリ36上の画像を表示するためのディスプレイ画面38と、CPU32に接続されたメモリ34と、時刻に関する情報をCPU32が得るためのタイマ40とを含んでいる。メモリ34は、複数のキーフレーム情報を記憶するキーフレーム情報記憶部42と、CPU32が実行する、本願発明のアニメーション表示方法を実現するためのプログラム情報を記憶するプログラム情報記憶部44とを含んでいる。

【0031】図2を参照して、キーフレーム情報記憶部42は、複数のキーフレーム情報62、64および66を含んでいる。なお通常キーフレームの数はこれよりも多いが、この例では説明を簡略にするためにキーフレームとして3つの場合のみを考えることとする。

【0032】たとえば第1のキーフレーム情報62は、第1のキーフレームが表示されるべき時刻を特定するための情報、すなわちアニメーションの表示開始から第1のキーフレーム表示までの予定経過時間72と、予定経過時間72の経過後にフレーム上に表示されるべき図形の特徴を特定するための図形情報74、78および82とを含んでいる。第1のキーフレーム62が表示されるのはアニメーションの開始時であるため、第1のキーフレームでは予定経過時間=0である。各図形情報74、78および82はそれぞれ、後述する第2のキーフレームの対応する図形が格納された領域へのポインタ76、80および84を有している。図形情報の内容については図5を参照して後述する。

【0033】同様にして第2のキーフレーム情報64は、予定経過時間92と、図形情報94、98、102および106とを含んでいる。第1のキーフレーム情報62の図形情報74、78および82のポインタ76、80および84はそれぞれ、第2のキーフレーム64の図形情報94、98および102の領域の先頭を指し示している。同様にして第2のキーフレーム情報64の図形情報94、98、102および106はポインタ96、100、104および108を有している。これらのうちポインタ96、100および108は、後述する

第3のキーフレーム66の対応する図形情報の領域の先頭を指し示している。第3の図形情報102のポインタ104は、キーフレームが終端であることを示す値を格納している。

【0034】同様にして第3のキーフレーム情報66は、予定経過時間112と、図形情報114、118および122とを含んでいる。第2のキーフレーム情報64のポインタ96、100および108はそれぞれ、この第3のキーフレーム情報66の図形情報114、118および122の領域の先頭を指し示している。図形情報114、118および122はそれぞれポインタ116、120および124を有しているが、本例ではこの第3のキーフレームが最終フレームであるものと仮定しているので、これらポインタはいずれも終端であることを示す値を格納している。

【0035】図2に示されるキーフレーム情報62、64および66が表わすアニメーションの概略を図3に示す。図3を参照して、このアニメーションは、画面130内を、三角形132が移動するものである。この三角形132は、図3に示されるように画面130の左上から中央上部に1秒かけて平行移動し、さらにそこから画面130の右下位置に1.5秒をかけて平行移動するものとする。図3において実線で示されている三角形は、それぞれ図2に示される第1のキーフレーム情報62、第2のキーフレーム情報64および第3のキーフレーム情報66によって特定される図形の形状および位置を示している。既に述べたように第1のキーフレーム62の予定経過時間は0である。第2のキーフレームで表示される三角形位置まで三角形132が移動する所要時間が1秒、さらにそこから第3のキーフレーム上で三角形132が表示される位置までの所要時間が1.5秒であるから、図2に示される第2のキーフレーム64の予定経過時間92は「1秒」であり、第3のキーフレーム66の予定経過時間112は「1+1.5=2.5秒」である。

【0036】図4(a)、(b)および(c)にそれぞれ、図2に示されるキーフレーム情報62、64および66によって特定されるキーフレームを示す。これら各キーフレームにおける三角形132の位置については既に述べたとおりである。

【0037】図5を参照して、図2に示されるキーフレーム情報62、64および66の内容について、説明する。この場合図3に示されるように、表示される図形は三角形132の1つであるから、図形情報としては1番目の図形情報74、94および114(図2参照)のみを想定する。図5(a)を参照して、第1のキーフレーム情報62の意味は次のとおりである。

【0038】「第1のキーフレームに対応する予定経過時間は0秒である。この第1のキーフレームの中には、図形情報74により特定される第1の図形が存在す

る。」

この第1の図形情報74の意味は、図5(a)に示される例では次のとおりである。

【0039】「第1の図形は、3個の点を持つ閉多角形である。すなわちこの図形は三角形である。三角形の3頂点の座標はそれぞれ(1, 2)(2, 3)(1, 4)である。この第1の図形に対応する図形は、第2のキーフレームの中にも存在しており、その図形情報は、図5(b)に示される第2のキーフレーム情報64内の第1の図形情報94である。」

同様にして図5(b)の第2のキーフレーム情報64の意味は次のとおりである。

【0040】「第2のキーフレームに対応する予定経過時間は1秒である。第2のキーフレームの中には第1の図形が存在し、その図形情報は図形情報94である。この図形情報94により特定される図形は、第1のキーフレーム情報62内の第1の図形情報74により特定される図形と連続している。図形情報94により特定される図形は、三角形であり、その3頂点の座標は(5, 2)(6, 3)(5, 4)である。第1の図形に対応する図形が第3のキーフレーム内にも存在しており、その位置はポインタ96により示されている。」

第3のキーフレーム情報66の意味は次のとおりである。

【0041】「第3のキーフレームに対応する予定経過時間は2.5秒である。第3のキーフレームには図形情報114により特定される図形が存在しており、その図形は三角形である。その三角形の3頂点の座標はそれぞれ(10, 6)(11, 7)および(10, 8)である。この図形は、第2のキーフレームの第1の図形から連続しているが、この第3のキーフレームで途切れている。」

図1に示されるプログラム情報記憶部44により記憶されており、CPU32により実行されることにより、特許請求の範囲に記載された各ステップの機能を実現するプログラムの制御の流れについて図6を参照して説明する。まずステップS1においてタイマ40(図1参照)が初期化される。本実施の形態では、初期化によりタイマ40の値が0にリセットされる。他の形態として、タイマ40は連続的に動作しており、この初期化時のタイマ40の値がメモリ34上の所定の領域に複写されて、以後の処理の基準値を示す値として利用されてもよい。

【0042】なお通常は、図6に示される処理に先立って、キーフレーム情報が、図1に示されないハードディスクなどの外部記憶装置からメモリ34のキーフレーム情報記憶部42内にロードされる。

【0043】続いてステップS2で、図5(a)に示される第1のキーフレーム情報62に基づき、1枚目のキーフレーム(図4(a))がグラフィックメモリ36(図1参照)上に描画される。ディスプレイ画面38

(図1参照)は、グラフィックメモリ36に書込まれた第1のキーフレームを表示する。

【0044】ところで、グラフィックメモリ36に図形を描画しその図形をディスプレイ画面38に描画する場合、描画対象となるメモリ領域と、表示中のデータが格納されているメモリ領域とを別にする方が、画面のちらつきを少なくできることが知られている。これを実現するためには2通りの方法が通常行なわれている。1つは、汎用メモリ上の領域をビットマップとみなしてここに図形を描画した後、描画された領域のデータをグラフィックメモリに高速に転送する方法である。他の1つは、グラフィックメモリを2画面分用意し、この2画面の双方に対して交互に図形を描画するとともに、描画が完了した領域の画面のみをディスプレイ画面に表示するという方法である。ここでは、いずれの方法を用いてもよい。どちらの方法を使用するにあたって、そのための特有の処理を行なう必要があるが、それらは当業者の知識の範囲内であるから、ここではその詳細については繰返さない。

【0045】次にステップS3で、キーフレーム番号を格納するための作業用レジスタiに初期値1を設定する。以後、各レジスタxの値を「変数x」と呼ぶこととする。ステップS4で、変数tに、ステップS1からの経過時間を格納する。図6に示されるように、ステップS1でタイマを初期化してから、ステップS2およびS3の処理を行なっているため、ステップS4時点での経過時間は0ではない。実際には、ステップS2およびS3のうち、ステップS2で行なわれる描画処理が最も複雑な処理であるから、経過時間の大部分をステップS2の処理時間が占めている。

【0046】以後、i番目のキーフレームの予定経過時間を P_i と呼ぶこととする。ステップS5で、i+1番目のキーフレーム(i=1であれば2番目のキーフレーム)の経過時間が変数t以下であるか否かについて判断を行なう。i=1の場合、 $P_{i+1}=P_2=1$ 秒である(図5(b))。この条件が成立すれば制御はステップS6に、成立しなければ制御はステップS8にそれぞれ進む。仮にステップS5の処理が最初に行なわれるとき(i=1)のtの値が0.3秒とする。この値と、図5(b)に示される第2のキーフレーム情報64内の予定経過時間の情報(=1秒)とが比較される。この場合には条件は成立しないので制御はステップS8に進むことになる。

【0047】ステップS8では、次に描画すべきフレームの位置づけを示す値hが次の式により計算される。

【0048】

【数1】

$$h = \frac{t - P_i}{P_{i+1} - P_i}$$

【0049】この場合の値hの示す位置を図解すると図

7のようになる。図7に示されるように、 $t=0.3$ は、時間軸上で、第1のキーフレームの表示される時刻と、第2のキーフレームが表示される時刻との間であって、第1のキーフレームが表示される時刻から第2のキーフレームが表示される時刻に向かって30%の位置に相当する。そこで値 h がこの t の位置づけを示しており、「次に描画すべき中間フレームの、時間軸上での位置づけ」を示しているものとする。

【0050】 $i=1$ の場合であって、かつ上述のように $t=0.3$ と仮定すると、 $P_i=0$ 、 $P_{i+1}=1$ であるから、 $h=0.3$ となる。

【0051】続いてステップS9で、変数 h で特定される時点に対応する中間フレームの各図形の各特徴値の座標が次のように計算される。これら各座標値を含めて i 番目のキーフレームのすべての図形のパラメータを Q_i と表わし、 $i+1$ 番目のキーフレームでの対応するパラメータを Q_{i+1} と表わすと、パラメータ Q_i とパラメータ Q_{i+1} に対応するパラメータを R とすると、 R は次のようにして計算される。

【0052】

【数2】

$$R = Q_i + h \times (Q_{i+1} - Q_i)$$

【0053】図3～図5に示される例を用いて具体的に示すと次のようになる。

【0054】

【表1】

座標1のX値	$1 + 0.3 \times (5 - 1) = 2.2$
座標1のY値	$2 + 0.3 \times (2 - 2) = 2$
座標2のX値	$2 + 0.3 \times (6 - 2) = 3.2$
座標2のY値	$3 + 0.3 \times (3 - 3) = 3$
座標3のX値	$1 + 0.3 \times (5 - 1) = 2.2$
座標3のY値	$4 + 0.3 \times (4 - 4) = 4$

【0055】すなわち、経過時間 $t=0.3$ (秒)に対応する中間フレームに描かれるべき三角形の各頂点の座標、すなわち座標1、座標2および座標3はそれぞれ(2.2, 2)(3.2, 3)および(2.2, 4)となる。

【0056】このようにしてすべてのパラメータ Q_i について R を求めて、この R の値を用いて中間フレームの全図形を描画する(ステップS10)。この全図形を描画が終了すると、処理はステップS4に戻る。すなわち、ある中間フレームの全図形を描画すると、その時点で次のフレームの描画を始めることとなり、その時点までの経過時間をステップS4で求めて以後の処理を繰返し実行する。

【0057】ところで、ステップS10で行なわれる図形描画にも一定の処理時間が必要とされる。この例では、図4(a)および(b)に示されるように、第1のキーフレームと第2のキーフレームとで、描くべき図形自体は同じ三角形132である。 $t=0.3$ の時点で描

くべき三角形も、たとえば図4(a)に示される初期状態($t=0$)での三角形132と同一の大きさである。この場合には $t=0.3$ の時点での図形の描画量自体は、 $t=0$ の時点と変わっていない。この中間フレームを描くために要する時間もまた、 $t=0$ のときの値とほぼ同じ0.3秒となるであろう。したがって、ステップS4の2回目の実行での経過時間 t は $t=0.6$ となる。

【0058】上述の繰返し処理において、ステップS5での判断がYESとなると処理はステップS6に進む。ステップS6では変数 i に1を加算する。さらにステップS7で、加算後の i を用いて $i+1$ が、当初用意されたキーフレームの数 n よりも大きいかなにかについての判断が行なわれる。この条件が成立しなければ制御はステップS5に進んで以下の処理を繰返し実行する。ステップS7の条件が成立すると、最後のキーフレームの予定経過時間を過ぎたということであるから、処理を終了する。

【0059】上記した例では、中間フレームで描画されるべき描画量は変化しない。したがって各中間フレームの描画に要する時間は0.3秒である。また $n=3$ である。したがってこの例で図6の処理を実行することにより、図8に示されるようなタイミングでフレームの描画が行なわれる。すなわちまず $t=0$ で第1のキーフレームが描画される。さらに $t=0.3$ 、 0.6 および 0.9 の時点で、第1のキーフレームと第2のキーフレームとの間での補間計算から得られた中間フレームが描画される。さらに $t=1.2$ 、 1.5 、 1.8 、 2.1 、および 2.4 の時点で、第2のキーフレームと第3のキーフレームとの間での補間計算により得られた中間フレームが描画される。すなわち各時点を中心として合計9枚のフレームが描かれ、連続したアニメーションとして表現される。

【0060】このときの画面130上に描かれる三角形132の軌跡を図9に、各図形の描かれる時点とともに示す。図9において、実線で示されているのがキーフレームの図形であり、破線で示されているのが中間フレームで描かれる図形である。図9に示される例で特徴となっているのは、キーフレームのうち最初のキーフレームは $t=0$ の時点で描かれているが、他のキーフレーム(第2および第3のキーフレーム)は描画されていないということである。

【0061】なお、図8および図9に示される $t=0 \sim 2.4$ の0.3秒おきの各時点は、描画の開始の時刻を示している。実際に画面に表示されるまでには、各フレームの描画に対する所要時間が必要であるから、その分だけの遅れが生じる。

【0062】上述の例では、アニメーション表示される三角形の大きさが変化せずただ移動のみする場合を示した。しかし、今までに示した装置およびプログラムを用

いて、キーフレーム間で三角形の大きさが変化するようなアニメーションを表示することもできる。この場合には、上述の例のように三角形の大きさが変化しない場合と比較して、三角形の面積が変化するために、描画量が変わり中間フレームの描画に要する時間が変わってくる。したがって、全く同じ装置およびプログラムを用いても、見かけ上の動作が上述の例とは異なってくる。

【0063】図10(a)～(c)に、このアニメーションのための3つのキーフレームを示す。図10(a)に示すように、画面130の左上に、三角形134が描かれるのが第1のキーフレームである。図10(b)を参照して、第1のキーフレームの三角形134が大きくなり(2倍)、三角形136として表示されるのが第2のキーフレームである。図10(c)を参照して、三角形136が画面左上から画面右上に移動した後の状態が第3のキーフレームである。

【0064】図11に、図10(a)～(c)に示される3つのキーフレームに対応するキーフレームデータ162、164および166を示す。図11を参照して、第1のキーフレームのためのキーフレーム情報162は、経過時間(=0秒)と、三角形134を示す第1の図形情報174とを含んでいる。図形の種類は閉多角形であり点の数は3、すなわち三角形である。この三角形の3頂点は(1, 2)(2, 3)(1, 4)という座標を有する。第1の図形情報174のポインタ176は、第2のキーフレームのキーフレーム情報164に含まれる第1の図形情報194の領域の先頭を示している。

【0065】第2のキーフレームのキーフレーム情報164は、経過時間(=1秒)と、第2のキーフレームでの三角形136に対応する図形情報194とを含んでいる。図形情報194によれば、描画の種類は閉多角形であり、その点の数は3、すなわち三角形である。三角形の3つの頂点の座標はそれぞれ(1, 1)(3, 3)(1, 5)である。ポインタ196は、第3のキーフレームのためのキーフレーム情報166の第1の図形情報214の領域の先頭を指し示している。

【0066】第3のキーフレームのためのキーフレーム情報166は、予定経過時間(=2.5秒)と、第1の図形のための図形情報214とを含んでいる。第1の図形情報214によれば、図形の処理は閉多角形であり、その頂点の数は3、すなわち三角形である。この三角形の3頂点の座標はそれぞれ(8, 1)(10, 3)(8, 5)である。第1の図形情報214のポインタ216は、この図形が第3のキーフレームで終端となっていることを示している。

【0067】図10(a)および(b)を参照して、この間の時間では、三角形134が徐々に拡大され三角形136となるアニメーションが生成される。この例では、三角形の外形を描くのみであるものとする、描くべき図形の大きさがa倍になると、その図形を描くため

の処理時間もほぼa倍となる。図10(a)(b)においては、描くべき図形は三角形134および136のみであるから、図10(b)に示される第2のキーフレームを描画するための処理時間は、図10(a)に示される第1のキーフレームを描画するための処理時間のほぼa倍となる。この例では $a=2$ であるから、第2のキーフレームの処理時間は第1のキーフレームのほぼ2倍である。但し前述したとおり、第2のキーフレームは実際には描画されない。

【0068】なお、上述の例では三角形の外形を描くのみであるが、三角形の内部を特定の色で塗り潰す場合には、縮尺がa倍になると面積は a^2 倍となるため、塗り潰しを行なうために必要な描画量がほぼ a^2 倍となる。したがって描画量の増加率は、図10(a)および(b)のように外形のみを描く場合よりも大きくなる。

【0069】図10(b)および(c)を参照して、第2のキーフレームから第3のキーフレームの間は、拡大された三角形が平行移動するアニメーションが行なわれる。この場合、1フレーム当りに必要な描画量は変化しないので、1フレーム当りの描画時間も変化せず、第2のキーフレームの処理時間と同様の処理時間を各フレームに対して必要とする。

【0070】すなわち、図10(a)～(c)に示されるようにキーフレームからアニメーションの中間フレームを生成してアニメーション再生した場合、フレームの描画開始タイミングは図12に示すようになる。まず第1のキーフレームの描画を開始する($t=0$)。第1のキーフレームの描画が、第1の例の場合と同様に0.3秒を要するものとする、第2のキーフレームの描画は $t=0.3$ の時点で開始される。図11に示される図形情報174および194によれば、第2のキーフレーム($t=1$ に相当)での三角形の大きさは、第1のキーフレームでの三角形134の大きさの2倍である。この間、三角形の拡大は線形的に行なわれるものとする、 $t=0.3$ の時点では、三角形134を1.3倍した大きさの三角形を描画する必要が生じる。この場合、 $0.3 \times 1.3 = 0.39$ (秒)、すなわち約0.4秒の処理時間を要する。 $t=0.3$ の時点で描画処理を開始した中間フレームの処理は $t=0.7$ の時点で終了する。さらに $t=0.7$ に相当する時点での描画されるべき三角形の大きさは、図10(a)の三角形134を1.7倍した大きさとなる。その描画に要する時間は $0.3 \times 1.7 = 0.51$ (秒)、すなわち約0.5秒である。したがってこの中間フレームの描画は $t=1.2$ 秒の時点で終了し、次のフレームの描画が開始される。 $t=1.2$ 秒の時点では描かれるべき三角形は既に図10(b)に示される三角形136の大きさとなり、これ以後その大きさには変化はない。したがって以後の描画に要する時間は一定である。この描画には、 $0.3 \times 2 = 0.6$ 秒を要する。このようにして各中間フレームの描

画が開始されるタイミングを示したのが図12である。こうして、図12に示されるように、 $t=0, 0.3, 0.7, 1.2, 1.8$ および2.4に相当する時刻を起点とした6枚のフレームが描かれて連続したアニメーションとして表現される。

【0071】図8と図12とを参照して、図8に示される例では各フレームを描画するための処理時間は一定であったのに対し、図12に示される例ではそれらは一定ではない。しかしいずれの場合にも、各キーフレームに保持されている予定経過時間情報に沿ってアニメーションの再生が行われ、かつ全体の再生時間も両者ともほぼ2.5秒となることがわかる。つまり、各フレーム当りの描画量が変化しても、その変化が全体のアニメーションの表示の進行には影響を与えないということがわかる。

【0072】同様の考え方によれば、たとえばCPUの処理能力が2倍になった場合を考える。図8に示す例の場合、CPUの処理能力が2倍になれば、各フレームの描画が開始されるタイミングは図13に示されるように0.15秒おきとなり、全体で17枚のフレームが再生される。また図10に示される例を同じく処理能力が2倍のCPUで再生した場合、各フレームの描画の開始タイミングは図14に示されるようになる。この場合、第1のキーフレームから第2のキーフレームまでの間に中間フレームの描画に要する時間は徐々に増加するが、全体としては10枚のフレームでこのアニメーションを再生することができる。図8と図13とを参照すれば、同じアニメーションが、同じ時間経過に沿って再生されることがわかる。しかもこの場合には、この間に生成されるフレーム数がほぼ2倍となり、より滑らかなアニメーションを得ることができる。図12と図14とを比較しても同様である。したがって、アニメーションの作成時にどのような処理能力のCPUを想定していたとしても、また再生時に実際にどのような処理能力のCPUでこのアニメーションを再生したとしても、この方法によりアニメーションを再生する限り、同じ時間経過に沿って、かつCPUの処理能力に応じた滑らかさでアニメーションが再生されることになる。

【0073】以上、図形の特徴点の座標が変化する場合について例とした示したが、同様の方法を用いて、データの中身を変えることにより、かつ図形を描画する場合に、データの内容に応じて異なる図形描画処理を行なうようにすれば、上述の実施の形態1のコンピュータおよびプログラムを用いて、図形の色、透明度、図形を構成する線の太さ、図形を構成する線が破線の場合の実線部と空白部との比率、図形が文字である場合のフォントのサイズ、図形が複数の図形からなる図形群の場合に、この図形群の基準点の座標、同じく図形群の表示上の縮尺比率などを、各キーフレームごとに指定しておき、中間フレームをこれらキーフレームの対応する情報の間で

行なう補間計算により得ることができる。

【0074】たとえば図15(a)に、図形の種類が多角形の場合に考えられる図形情報262のデータ構造を示す。図15(b)に、図形の種類として一群の文字からなるテキストが指定された場合の図形情報264のデータ構造の例を示す。図15(c)には、図形の種類として図形群が指定された場合の図形情報266のデータ構造の例を示す。

【0075】図15(a)を参照して、図形の種類が多角形の場合には、点の数と、点の数に応じた数の座標と、図形内部を塗り潰すべき色と、透明度と、図形を構成する線の太さと、その線種と、線種が破線である場合の実線部と空白部との比率を示すデータ等が格納される。たとえば色であれば、光の三原色である赤(R)、緑(G)、青(B)の数値により表現できる。連続した2つのキーフレーム中の図形の属性情報として、この(R、G、B)を指定しておけば、対応するデータの間の補間計算を行なって中間フレームの図形を塗り潰すべき色を特定する(R、G、B)を計算することができる。透明度、線の太さ、線種が破線の場合の実線部/空白部の比率などについても全く同様である。

【0076】図15(b)を参照して、図形の種類がテキストである場合には、そのテキストを描画すべき座標(この場合にはテキストの基準位置の座標)と、テキスト内容と、テキストを表示すべきフォントの種類と、テキストを表示すべきフォントのサイズを示すフォントポイント数と、テキスト表示の色を指定する情報と、フォントを表示する際のそのフォントのスタイル(標準、イタリック体、ゴールド体など)と、テキスト中の文字間隔、行間隔、各行についての行揃えを示す情報、などが格納される。特に文字の場合、連続した2つのキーフレーム中のテキストの属性情報として、ポイントフォント数があれば、その値に対して補間計算を行なうことにより中間フレームで表示すべきテキストの文字の大きさを決定することができる。この場合、ポイント数が端数であってもそのポイント数に従った大きさの文字を表示できる場合には問題ないが、特定のポイント数の文字しか表示できない場合には、補間によって得られたポイント数に最も近いポイント数の文字を表示するようにしてもよい。

【0077】図15(c)を参照して、図形情報266の種類の図形群の場合には、この情報266には、その図形群に含まれる図形の数と、各図形に対応する図形情報への、その図形の数に対応したポイントと、図形群全体を表示する際の基準点の座標と、図形群を表示する際の表示上の縮尺比率などが格納される。この場合のポイント数は、この図形群の図形情報266が格納されているキーフレームの中に別の図形情報として格納されている個々の図形情報へのポイントである。図15(c)に示される例の場合、基準点座標、表示上の縮尺比率など

を、隣接する2つのキーフレーム間の補間計算により求めることができる。

【0078】次に、本発明の実施の形態2に係るアニメーション表示方法を実現するための装置について図16を参照して説明する。実施の形態1では、隣接する2つのキーフレーム間で対応する図形情報のうち対応する特徴データの間で線形計算による補間計算で中間フレームの図形情報の対応する特徴データを生成していた。実施の形態2の方法では、このようにして補間計算により得られた特徴データをさらに所定の方法で変換して、時間軸上での各図形の動きに、線形計算では得られない変化を与えることを特徴としている。

【0079】図16に、実施の形態2の方法を実現するためのコンピュータ330のブロック図を示す。このコンピュータ330が図1に示される実施の形態1を実現するためのコンピュータ30と比較して異なるのは、メモリ34内に、実施の形態1のためのプログラムとは異なるプログラムを記憶したプログラム情報記憶部344と、上述のように一旦線形計算による補間計算により得られた図形の特徴データをさらに変換する際に用いられる変換テーブル346を含んでいる点である。その他の点では図16に示されるコンピュータ330は図1に示されるコンピュータ30と同様である。同一の部品には同一の参照符号および名称を与えており、それらの機能も同一である。したがってここではそれらについての詳細は繰返さない。

【0080】図17を参照して、変換テーブル346は、実施の形態1で説明された、「次に描画すべき中間フレームの位置づけ」を示す値 h を予め計算し、その h の各値に対して、その時点で実際の変化量としてとるべき値 k を予め定め、 h と k との対をテーブルとして予め準備したものである。図17に示される変換テーブル346では、 h に対して $k = h^2$ の関係が想定されている。なお、 k として h と上述したような関数関係を想定して変換テーブル346を作成してもよいが、変換テーブルを用いた場合には、必ずしもそうした関数関係に基づく、アニメーション作成者の恣意的な意図によって k を定めることができ、それによって単純な関数では得られないような特殊な動きを実現することが可能になる。

【0081】図27を参照して、このような変換テーブルを複数個準備し、それぞれの図形情報に付加されたテーブル番号によって、それぞれ異なる変換テーブルを対応付けることにより、キーフレームが2つだけであっても、それぞれの図形に、時間軸に対して異なる動きを表現させることが可能になる。なお、ここではテーブル番号=0は、変換テーブルを用いないということを表すものである。

【0082】また、図28を参照して、テーブル番号1は k の変化量が次第に増加する加速度運動、テーブル番

号2は k の値が0と1の間で振動する反復運動を表している。図29を参照して、等速で移動する飛行機の絵1001には変換テーブルを対応させないものとし、落下するリングの絵1002にはテーブル番号1すなわち加速度運動を対応させ、しゃべる人物の絵1003にはテーブル番号2すなわち反復運動を対応させるものとする。キーフレーム1とキーフレーム2の2つのキーフレームだけで、3種類の異なる性質の動きを同時に実現することが可能となる。

【0083】図18に、図16に示すプログラム情報記憶部344に格納されるプログラムの制御の流れを示す。図18に示すフローチャートが図6に示される実施の形態1のプログラムのフローチャートと異なるのは、ステップS8の後に、ステップS8で得られた h を、変換テーブル346を参照して k に変換するステップS11が含まれることと、図6のステップS9で h の代わりに k を用いてステップS9と同様の処理を行なうステップS12が挿入されていることとである。他の点では図18は図6に示されるフローチャートと全く同様である。

【0084】このように、線形計算による補間計算により得られた値 h を、変換テーブル346を用いて k に変換し k を用いて中間フレームの対応パラメータを計算して図形を生成する。こうした処理をたとえば図形の座標点に対して適用すると、たとえば加速度を伴った図形の動きなど、時間の進行に対して非直線の動きを表現することが可能になる。また、既に述べたようにテーブル346の内容を自由に設定できることから、時間に対して恣意的な動きを設定することができるという利点がある。また図形の色に対して適用すると、時間軸上で色が変化する特殊な効果を出すことができる。

【0085】このように一旦 h を計算した後、 h を別の値 k に変換する方法としては、テーブル変換ではなく或る関数の変数として h を代入し、そのときの関数の値を k とすることも考えられる。たとえば $k = h^2$ という演算を行なうと k を得ることが考えられる。もちろんこの例は一例であり、他のどのような関数を用いてもよい。この場合、図18のステップS11で、上述の関数による変換を行なえばよい。この場合には、テーブル変換による場合と比較して、保有しなければならないデータ量がより少なくなるという利点がある。

【0086】上述のように本発明によれば、隣接するキーフレームの対応する図形の対応するパラメータ間で補間を行なうことにより中間フレームを作成するキーフレーム補間方式のコンピュータアニメーションにおいて、CPU速度や1フレーム当たり必要とされる描画量に依存せず、全体の再生時間を一定とすることが可能となる。そのため画像は、音声と同期して再生させることが容易となる。また、CPUの処理能力に応じて増加する中間フレーム数でアニメーションを再生することができ、か

つCPUに遊び時間が発生することがないため、CPUの処理能力の増大に応じ過去に作成したデータを修正することなく、より滑らかなアニメーションを再生することができる。

【0087】本発明におけるアニメーション表示プログラムは、磁気テープ又はCD-ROM等の記録媒体において供給される。当該アニメーション表示プログラムはコンピュータ本体によって実行され、操作者はグラフィックディスプレイ装置を見ながらキーボードまたはマウスを操作することによってアニメーション表示を行う。また、アニメーション表示プログラムは他のコンピュータより通信回路を経由し、通信モデムを介してコンピュータ本体に供給されてもよい。

【図面の簡単な説明】

【図1】本発明を実現するためのコンピュータのブロック図である。

【図2】キーフレーム情報を模式的に示す図である。

【図3】アニメーションの一例を模式的に示す図である。

【図4】図3に示されるアニメーションを再生する上で必要とされるキーフレームの画像を示す図である。

【図5】図4に示されるキーフレームのためのキーフレーム情報を示す図である。

【図6】実施の形態1でのアニメーション表示方法を実現するためのプログラムのフローチャートである。

【図7】中間フレームの生成タイミングを模式的に示す図である。

【図8】描画量が一定の場合の、中間フレームの描画開始タイミングを模式的に示す図である。

【図9】図3に示されるアニメーションを再生する場合の、中間フレームの各図形を模式的に示す図である。

【図10】図形の描画量が変化するアニメーションのためのキーフレームを示す図である。

【図11】図10に示されるアニメーションのキーフレーム情報を示す図である。

【図12】図10に示されるキーフレームから再生されるアニメーションの、各フレームの描画開始タイミングを模式的に示す図である。

【図13】図3に示されるアニメーションを、処理能力の高いCPUで処理した場合の、各フレームの描画開始タイミングを模式的に示す図である。

【図14】図10に示されるアニメーションを、処理能力の高いCPUで処理した場合の、各フレームの描画開始

始タイミングを示す模式図である。

【図15】図形の色など、座標情報以外のデータを格納した図形情報の構成の例を示す図である。

【図16】本発明の実施の形態2を実現するためのコンピュータのブロック図である。

【図17】変換テーブルを模式的に示す図である。

【図18】実施の形態2の方法を実現するためのプログラムのフローチャートである。

【図19】従来のキーフレーム補間方式のアニメーションの原理を模式的に示す図である。

【図20】従来方式によるアニメーション表示の一例を示す図である。

【図21】従来技術によるキーフレーム補間方式のアニメーションの問題点を示すための図である。

【図22】従来技術によるキーフレーム補間方式のアニメーションの問題点を示すための図である。

【図23】従来技術によるキーフレーム補間方式のアニメーションの問題点を示すための図である。

【図24】従来技術によるキーフレーム補間方式のアニメーションの問題点を示すための図である。

【図25】従来技術によるキーフレーム補間方式のアニメーションの問題点を示すための図である。

【図26】従来技術によるキーフレーム補間方式のアニメーションの問題点を示すための図である。

【図27】本発明の実施の形態2における複数の変換テーブルを持つ場合のキーフレーム情報を模式的に示すための図である。

【図28】本発明の実施の形態2における複数の変換テーブルの例を示すための図である。

【図29】本発明の実施の形態2における複数の変換テーブルを利用したアニメーションのキーフレームの例を示すための図である。

【符号の説明】

30、330 コンピュータ

32 CPU

34 メモリ

36 グラフィックメモリ

38 ディスプレイ画面

40 タイマ

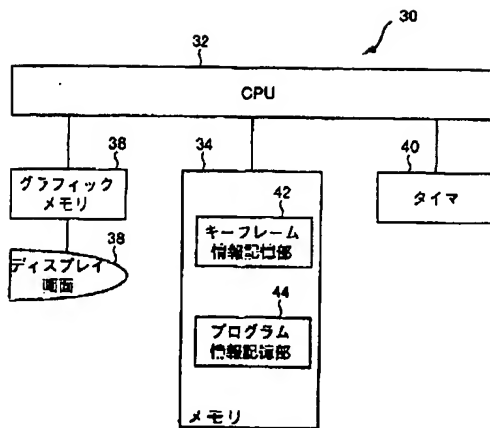
62、64、66、162、164、166 キーフレーム情報

346 変換テーブル

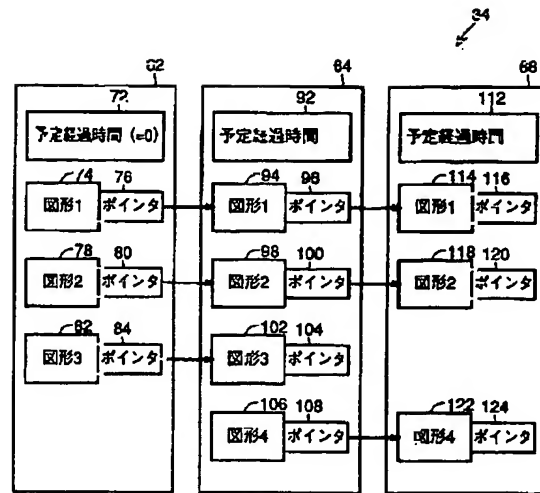
【図28】

テーブル番号1 (加減成算)	h	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
	k	0	0.01	0.04	0.09	0.16	0.25	0.36	0.49	0.64	0.81	1.0
テーブル番号2 (乗算)	h	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
	k	0	0.5	1.0	0.5	0	0.5	1.0	0.5	0	0.5	1.0

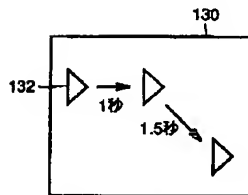
【図1】



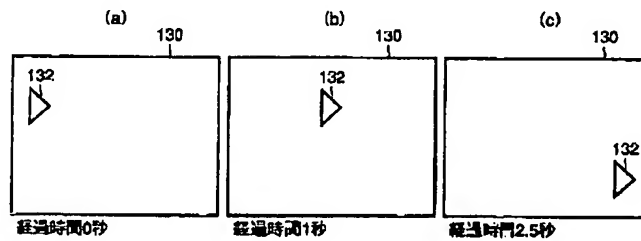
【図2】



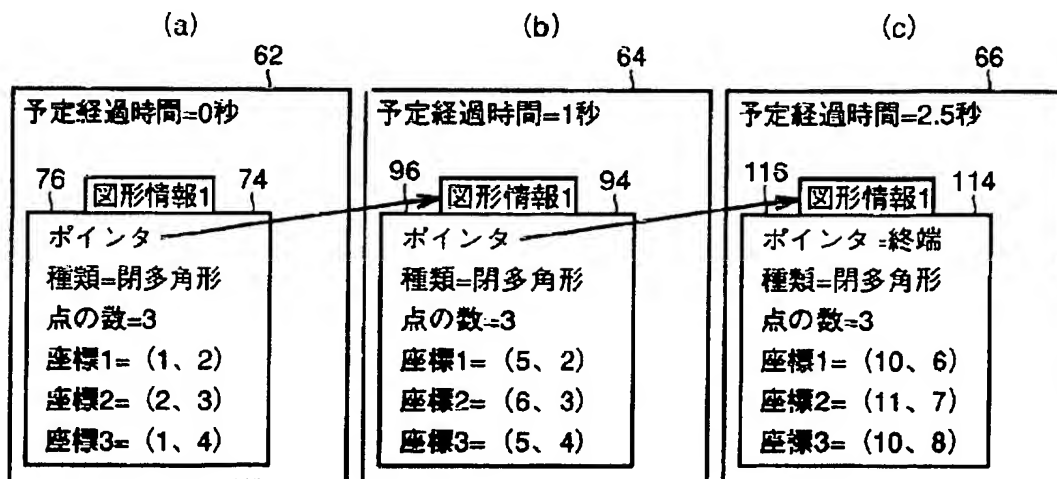
【図3】



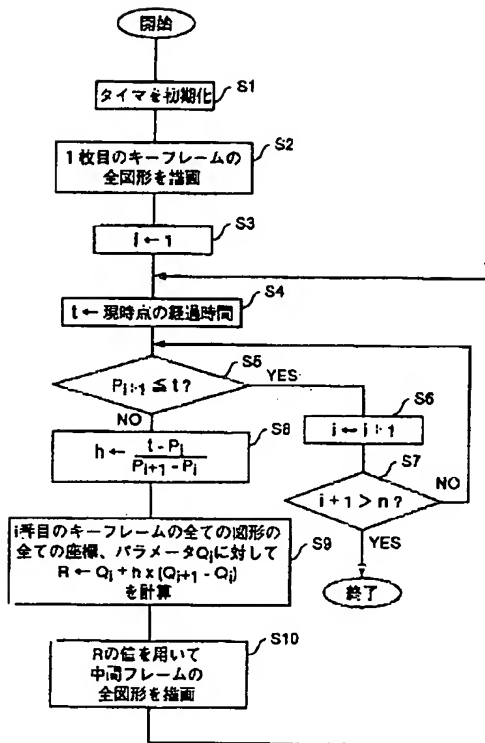
【図4】



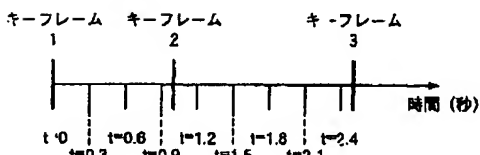
【図5】



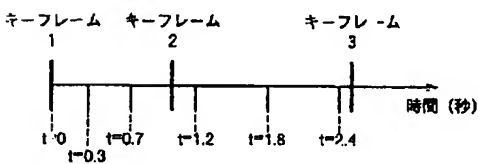
【図6】



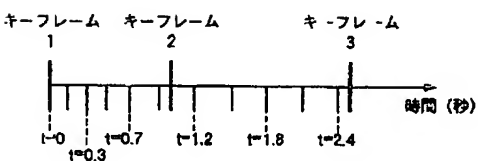
【図8】



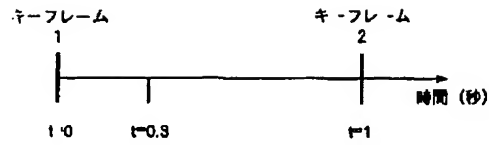
【図12】



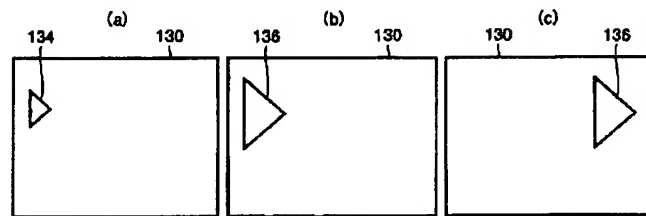
【図14】



【図7】



【図10】

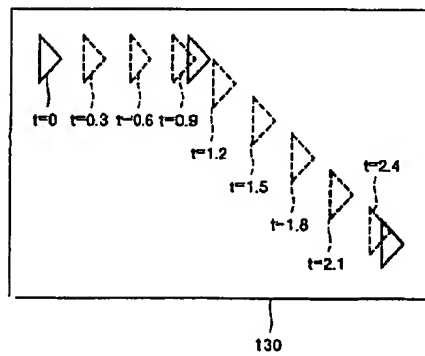


【図17】

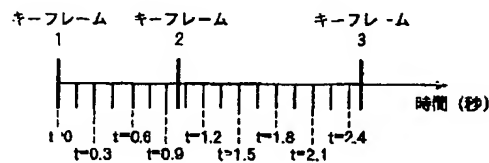
346

h	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
k	0	0.01	0.04	0.09	0.16	0.25	0.36	0.49	0.64	0.81	1.0

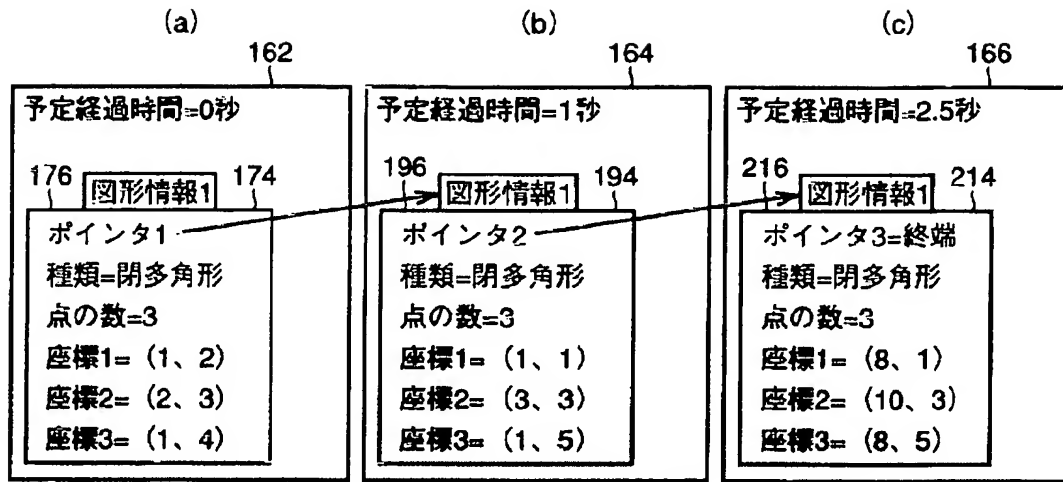
【図9】



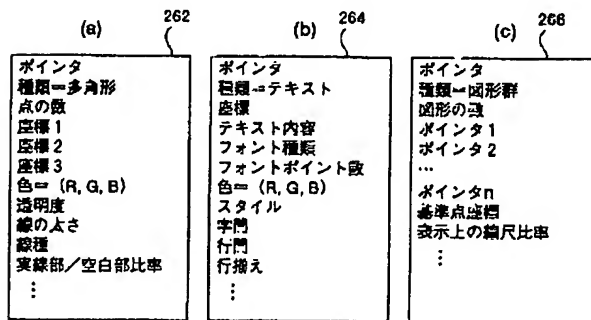
【図13】



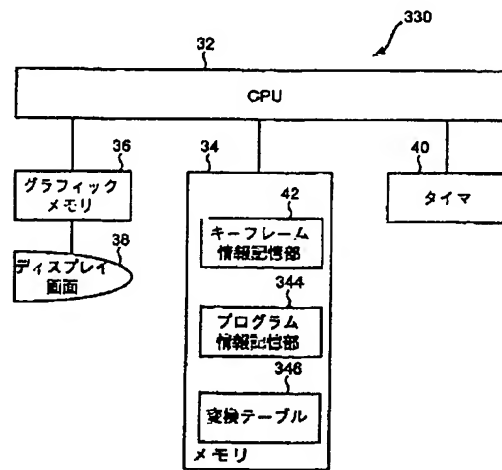
【図11】



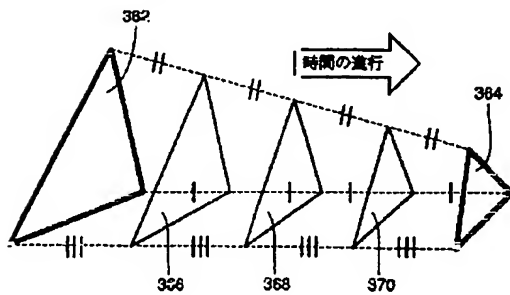
【図15】



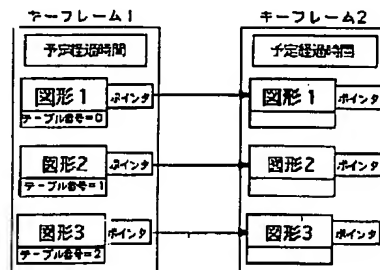
【図16】



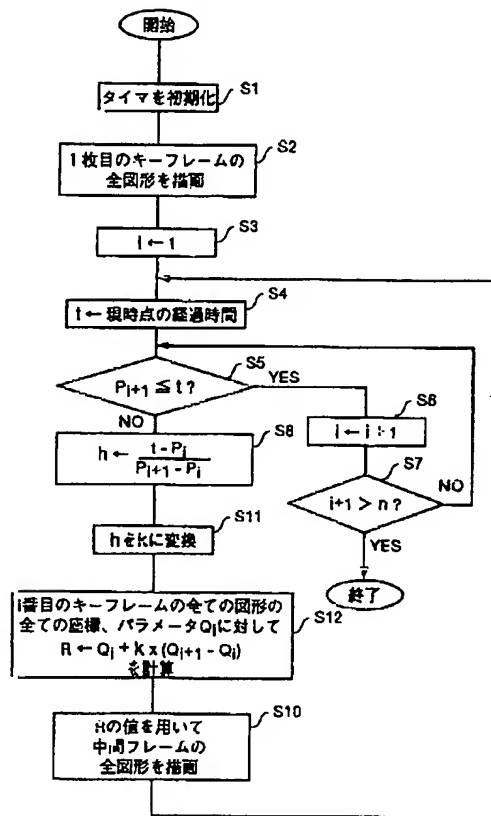
【図19】



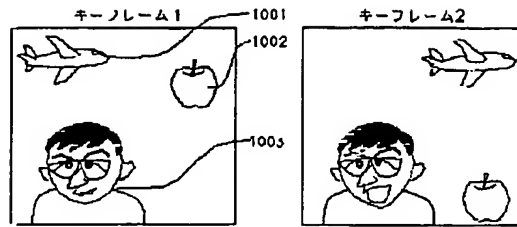
【図27】



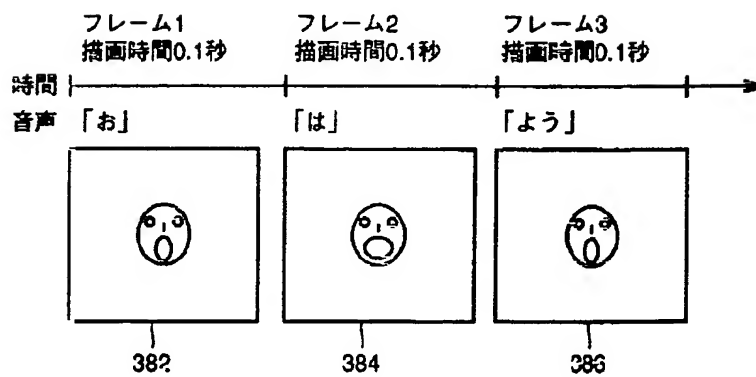
【図18】



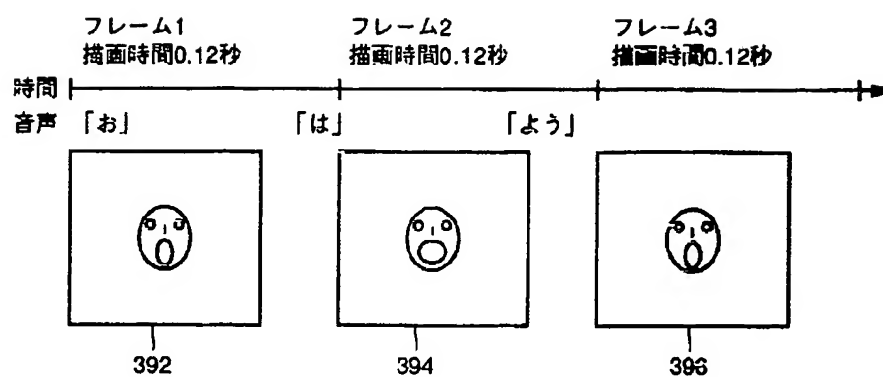
【図29】



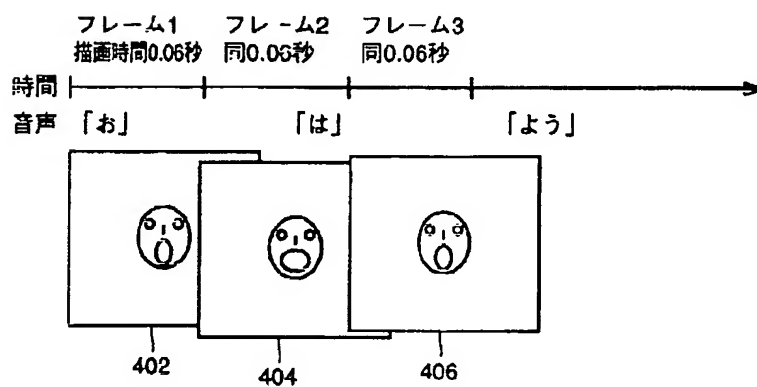
【図20】



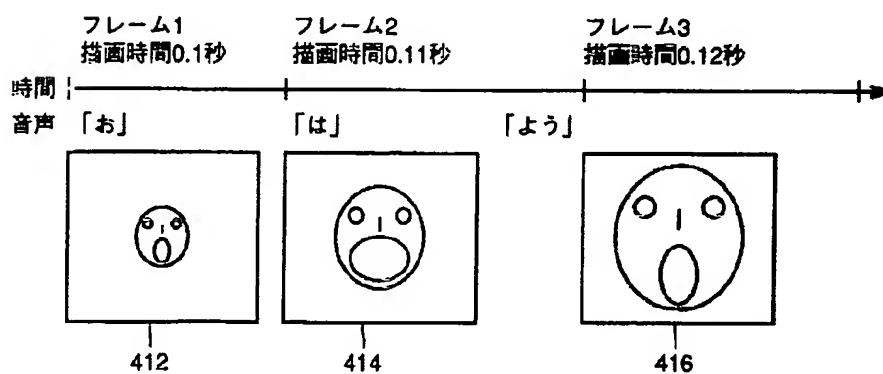
【図21】



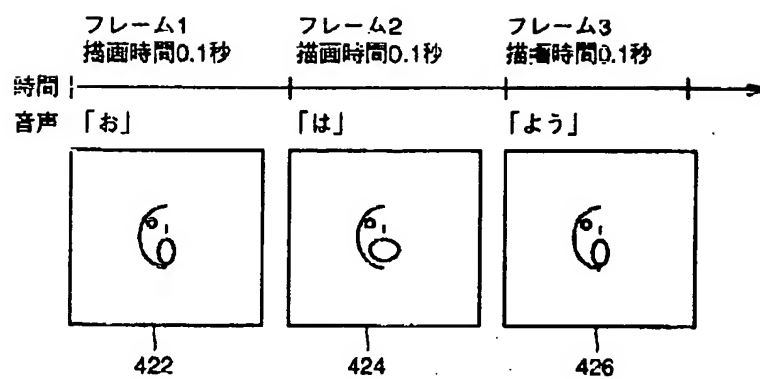
【図22】



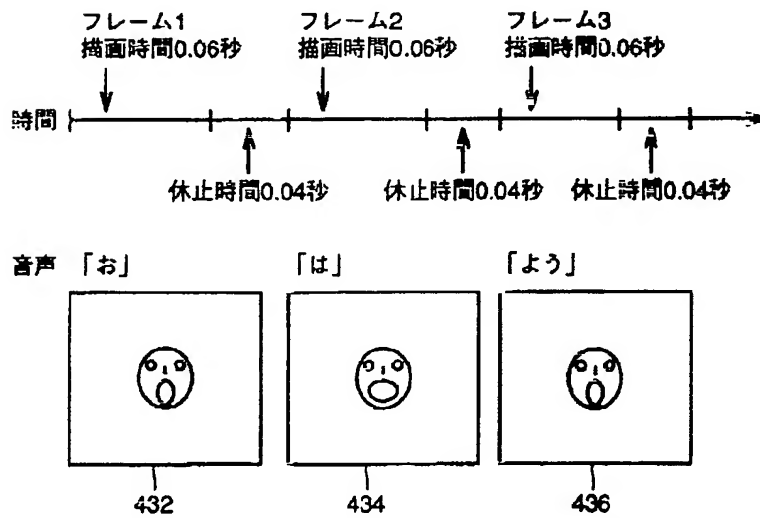
【図23】



【図24】



【図25】



【図26】

